

VORTEX I/VORTEX II REFERENCE CARD

Parameter Legend:

- **Boldface** = Obligatory
- *Italic* = Optional
- UPPER CASE = AS SHOWN
- Lower Case = A variable

OPERATOR COMMUNICATION

Schedule Foreground Task
;SCHED,task,level,lun,key

Time-Schedule Foreground Task
;TSCHED, task,level,lun,key,time

Attach Foreground Task
;ATTACH,task,line,iew,enable
iew interrupt event word

enable E = enable
D = disable

Resume Task
;RESUME,task

Task Status
;TSTAT,task

Task Status (TIDB Words 1 and 2)

TIDB Word	Bit	Meaning of Set Bit
1	15	Suspend interrupt
1	14	Suspend task
1	13	Abort task
1	12	Exit from task
1	11	TIDB resident
1	10	Resident task
1	9	Foreground task
1	8	Protected task
1	7	Task scheduled by time-delay
1	6	Time-delay active
1	5	Task waiting to be loaded (check pointed)
1	4	Task error
1	3	Task interrupt expected
1	2	Overlay task
1	1	Scheduled task upon termination of active task
1	0	Task search-allocated-loaded
2	15	Task opened, but not loaded
2	14	V75 (long TIDB)
2	13	Load overlay
2	12	Background checkpoint I/O wait
2	11	Allocation override flag
2	10	Background being checkpointed
2	9	TIDB not available
2	8	map key assigned
2	7	map key checkpointed
2	6	Delay type 3 request
2	5-0	Task priority level

②

Enter/Display Time-Of-Day
;TIME,time
HH:MM

Enter Date
;DATE,mm/dd/yy

Abort Task
;ABORT,task

Assign Logical Unit(s)
;ASSIGN,l(1),=r(1),l(2),...,l(n)=r(n)

Device Down
;DEVDN,device

Device Up
;DEVUP,device

List Logical-Unit Assignments
;LOLIST,lun(1),lun(2),...,lun(n)

BLDSKD
;SCHED,BLDSKD,5,FL,F

(RESPONSE) ENTER ACTION LETTER A, D or L
A = Add (a new Task)
D = Delete (an existing Task)
L = List (all Task letters)

(A-RESPONSE)
DIRECTIVE FORMAT: LETTER,TASK,LIB,KEY,PRI
LETTER = Alpha portion of control
TASK = Task name to be scheduled
LIB = Lib where Task resides
KEY = KEY Protect Key
PRI = Priority for schedule

JOB CONTROL PROCESSOR

Global-Lun assign and open
/AFILE,L(1),L(2),Key,Filename

L(1) = is the global Lun name or #
L(2) = is the Lun name or # of
RMD to be assigned

Schedule Next Background Task
/ALTLIB,lun,key
must be followed by a /LOAD or /PLOAD

Logical-Unit Assignment
/ASSIGN,l(1),i(2)=r(2),...,i(n)=r(n)
l(n) lun or name
r(n) lun, name, or physical device name

Comment
/C,comment

Close File (RMD/MT)
/CFILE,lun,key,name
lun must be a global logical unit
name file name closed with update

③

(4)

*Close VTAM Line/Terminal
/CLOSE,lun,id

COBOL Compiler

/COBOL,p(1),p(2),...,p(n)

Parameter

B suppresses binary object
O object listing
L outputs binary object on GO file
M suppresses data division map
N suppresses source listing

Compression Edit System

/COMSY

System Concordance

/CONC,L

L list all symbols

DAS MR Assembler

/DASMR,p(1),p(2),...,p(n)

Parameter

B suppresses binary object
E multiple-register (V75) instructions
I flags implicit indirect instructions
L outputs binary object on GO file
M suppresses symbol table listing
N suppresses source listing

Data Base Generator

/DBGEN

Dump Background

/DUMP

must be followed by a /LOAD

Execute

/EXEC,D

D dumps all of the background upon completion of execution

Terminate Job

/ENDJOB

Terminate Operation

/FINI

terminates all JCP background operations

File Maintenance Task

/FMAIN

File Maintenance Utility

/FMUTIL

Set Line Count

/FORM,lines

FORTTRAN Compiler

/FORT,p(1),(2),...,p(n)

Parameter

A produce "DASMR" listing
B suppresses binary object
D assigns two words to integers and arrays
H generates code for FPP calls

(5)

L outputs binary object on GO file
M suppresses symbol table listing
N suppresses source listing
O outputs object module listing
X compiles conditionally
F generates code for firmware calls

I/O Utility Processor

/IOUTIL

Start Job

/JOB,name

Set Keypunch Mode

/KPMODE,m

m 0=026 mode
1=029

Load Module Generator

/LMGEN,M,N

M suppresses output of a memory map
N print alpha and numerical sorted map

Load/Execute Background Task

/LOAD,name,p(1),p(2),...,p(3)

name task name
p(1) user parameters

Allocate Extra Memory

/MEM,n

*Open VTAM Line/Terminal

/OPEN,Lun1,id,lun2

Pause

/P

Position RMD File

/PFILE,lun,key,name

lun number or name of the logical unit
must be a system defined logical unit which as a global FCB.
name file name

Schedule Background Library Task

label /PLOAD,name,p(1),p(2),...,p(n)

name task name
p(1) user parameter

Rewind

/REW,lun,lun,...,lun

RPG II Compiler

/PRG,p(1),p(2),...,p(n)

Parameter

B suppresses binary object
O include RPG debug features in object module
L outputs binary object on GO file
M suppresses symbol table listing
N suppresses source listing

Symbolic Source-Editor Task

/SEDT

(6)

Skip File(s)

/SFILE,lun,neof

System Maintenance Task

/SMAIN

Skip Record(s)

/SREC,lun,nrec,direc
direc F = forward
R = reverse

Set External (RPG II) Indicators (U1-U8)

/SWITCH,aaaaaaaa
a= 1 for switch ON
0 for switch OFF
x for no change

Write End-of-File Mark

/WEOF,lun

* = VTAM only

FILE MAINTENANCE

/FMAIN (JCP Directives)

FM**

Add

ADD,lun,key
lun output logical unit
add uses "create" function

Create

CREATE,lun,key,name,words,records
name filename
words words in each record
records number of records

Delete

DELETE,lun,key,name

Enter New File Name

ENTER,lun,key,old,new

Initialize

INIT,lun,key

Input

INPUT,lun,key,file
file name of the RMD file

Note: INPUT must precede ADD

List

LIST,lun,key

Rename

RENAME,lun,key,old,new

5

- L outputs binary object on GO file
- M suppresses symbol table listing
- N suppresses source listing
- O outputs object module listing
- X compiles conditionally
- F generates code for firmware calls

I/O Utility Processor

/IOUTIL

Start Job

/JOB,name

Set Keypunch Mode

/KPMODE,m

m 0=026 mode
1=029

Load Module Generator

/LMGEN,M,N

M suppresses output of a memory map
N print alpha and numerical sorted map

Load/Execute Background Task

/LOAD,name,p(1),p(2),...,p(3)

name task name
p(1) user parameters

Allocate Extra Memory

/MEM,n

*Open VTAM Line/Terminal

/OPEN,Lum1,id,lun2

Pause

/P

Position RMD File

/PFILE,lun,key,name

lun number or name of the logical unit
must be a system defined logical
unit which as a global FCB.
name file name

Schedule Background Library Task

label /PLOAD,name,p(1),p(2),...,p(n)

name task name
p(1) user parameter

Rewind

/REW,lun,lun,...,lun

RPG II Compiler

/PRG,p(1),p(2),...,p(n)

Parameter

- B suppresses binary object
- O include RPG debug features
in object module
- L outputs binary object on GO file
- M suppresses symbol table listing
- N suppresses source listing

Symbolic Source-Editor Task

/SEDIT

6

Skip File(s)

/SFILE,lun,neof

System Maintenance Task

/SMAIN

Skip Record(s)

/SREC,lun,nrec,direc

direc F = forward
R = reverse

Set External (RPG II) Indicators (U1-U8)

/SWITCH,aaaaaaaa

a= 1 for switch ON
0 for switch OFF
x for no change

Write End-of-File Mark

/WEOF,lun

* = VTAM only

FILE MAINTENANCE

/FMAIN (JCP Directives)

FM**

Add

ADD,lun,key
lun output logical unit
add uses "create" function

Create

CREATE,lun,key,name,words,records
name filename
words words in each record
records number of records

Delete

DELETE,lun,key,name

Enter New File Name

ENTER,lun,key,old,new

Initialize

INIT,lun,key

Input

INPUT,lun,key,file
file name of the RMD file

Note: INPUT must precede ADD

List

LIST,lun,key

Rename

RENAME,lun,key,old,new

7

FILE MAINTENANCE UTILITY

/FMUTIL (JCP Directive)

FU**

Dump File

D,lun,key,file,tapelun
file All (dump partition)
DIR (dump filename directory)
filename

Load File

L,lun,key,file,tapelun
file ALL (load partition)
DIR (load filename directory)
filename

Rewind

R,lun

End of File

E,lun

Search

S,lun,key,tapelun

Prints

P,lun,key

Release Unused Space

U,lun,key,file

Exit FMUTIL

E

LOAD MODULE GENERATOR

/LMGEN,M,N (JCP Directive)

LM**

Load Relocatable Object Modules without Re-opening or Repositioning

CLD,lun,key,file
key and file for RMD only

Load Relocatable Object Modules

LD,lunkkey,file
file filename

Library Search

LIB,lun(1),lun(2),key(2),...,lun(n),key(n)

Memory

Mem,n

End LMGEN

END,lun,key

Overlay

OV,segname

Create TAsk-Identification Block

TIDB, name type, segments, *DEBUG, ropages*

name task name

type 1=background task

2=foreground task

3=background task in alternate library

segments number of overlay segments

ropages read only page specifier

RELINK

Reestablish VORTEX nucleus pointers

/LOAD,RELINK

(response) LUN,KEY,LD MODULES

LUN = name or number where load module resides

KEY = protect key if any

mod name = up to ten mods (or a continuation,
to replace lun and key followed by more mod
names on the same lun

END = to exit from RELINK

INPUT/OUTPUT UTILITY

/IOUTIL (JCP Directive)

IU**

Close File

CFILE,lun,key,name,add

add 0=unchanged

1=updated

Copy File

COPYF, f, iu, im, irl, ou (1), om, orl, ou (2), ou (3),...ou (n)

im/om 0=binary

1=ASCII

2=BCD

3=unformatted

Copy Record

COPYR, r, iu, im, irl, ou (1), om, orl, ou (2), ou (3),...ou (n)

Format and Dump

DUMP, r, iu, im, irl, ou

Pack Binary

PACKB, f, iu, im, irl, ou (1), om, orl, ou (2),...ou (n)

Position File

PFILE, lun, key, recl, name

Print File

PRNTF, f, iu, ou (1), ou (2),...ou (n)

Rewind

REW, lun, lun,...,lun

Skip File

SFILE, lun, neof

Skip Record

SREC, lun, nrec

Write End Of File

WEOF, lun, lun,...,lun

VSORT (SORT/MERGE)

/LOAD, VSORT (JCP Directive)

SORT

INPUT = (lun, filename, key, record length)

OUTPUT = (lun, filename, key, record length)

$$\text{WORK} \left\{ \begin{array}{l} 1 \\ 2 \\ 3 \end{array} \right\} = (\text{lun, filename, key})$$

SORT KEY = (sc(1), ec(1), order (1),...,sc(29), ec(29),order (29))

sc (1) starting character (or byte) position

ec (1) ending character (or byte) position

order (1) A=ascending

D=descending

ENDSORT

DEBUGGING AIDS

DG** (TIDB task name, task entry address)

Directive Description

A Display and change the contents

Ax Change, but do not display, the contents

B Display and change the contents

Bx Change, but do not display, the contents

Cx Display and change the contents of memory
address x

Fx Set data address to value of x

Gx Load the contents of the pseudoregisters
into the respective registers, and transfer
to memory address xlx,y,z Initialize memory addresses x through y
with the value of z

O Display and change the overflow indicator

P Read DEBUG directives from PI unit until EOF

QC Clear all traps and restore original contents to trap cells

QL Display current trap locations

QS,v,w,x,y,z Place traps at memory addresses v,w,x,y,z

*Rn Display and change V75 registers

*Rnx Change, but do not display V75 Registers

Sx,y,z,m Search memory addresses x through y for the
z value, using mask mTy, X Place a trap at memory address y, starting execution
at address xTy Place a trap at memory address y, starting execution
at the last trap location

X Display and change the contents

Xy Change, but do not display, the contents

xxxxxx Display the contents of memory address xxxxxx

xxxxxx,

yyyyyy Display the contents of memory addresses xxxxxx
through yyyyyy

* Extended Instruction Set

SNAP

EXT SNAP

CALL SNAP

DATA start (start of display)

DATA end (end of display)

DATA tidb is less than zero if dump of task
TIDB is desired, is positive if TIDB dump
is to be suppressed

SN** (task TIDB name)

PATCH

;Sched, PATCH, 5, FL, F

PT**

.PTCH Modify main memory nucleus

.DUMP Create patch image file
(PTCHIM)

.APND Add to end of image file

.HIST List current PTCHIM

.BASE Set an address to a
specific character symbol

.LIBR modify a load module

(response) LUN, KEY, FILENAME, OVLY

.EXIT

CHANGE DIRECTIVES DO NOT START WITH
A PERIOD - FORMAT rs:

addr,value (1), value (2)...value(n)

SYSTEM MEMORY DUMP

/LOAD,DSPMEM

FIL, name

name = image file name to be examined

TID, name

name = TIDB name (or octal addr.) or
'ALL' for all TIDBs to be
displayed

TSK, name, p(1)...p(n)

name = Task TIDB name
p(n) = dump limit parameters

P = program region
only

T = Table region only

F = Foreground common
only

M = No dump of
mapped-in pages

Sxxx = Start page #
(dec. or octal)

Exxx = End page #

V = Exclude dump
of V.N.O. Tasks

blank = dump entire
Task

END

SOURCE EDITOR

/SEDIT (JCP Directive)

SE**

Add Record(s)

Add, recno

recno is the number of the record last copied
from the IN unit before switching to
AL for further copying.

Assign Logical Units

AS, nn=lun, key, file
nn IN, OU, AL
file filename

Compare Inputs

CO, (first, last), limit
limit discrepancy count allowed

Delete Record(s)

DE, recno 1, recno 2

Copy File(s)

FC, nfiles

Gang Load All Records

GA, (first, last), 'string'

List Records

LI, list

list A = list all out records
C = list only change records
N = suppress listing, except directives

Move Record(s)

MO, recno 1, recno 2, recno 3

Replace Record(s)

REPL, recno 1, recno 2

Rewind

REWl, p(1), p(2), p(3)
p(1) = IN, OUT, ALT

Write End of File

WE

Add String

SA, recno, (first, last), 'string'

Start Sequencing

SE, (first, last), initial, increment

Delete String

SD, recno, (n1, n2, n3)

Sequence Records

SE, (first, last), initial, increment

Stop Sequencing

SE, N

Replace String

SR, recno, (n1, n2, n3), 'string'

COMPRESSION/EDIT SYSTEM

/COMSY

(No RMD) ASSIGN, unit, Lun, R
unit = allowable SI,PI,BO,LO and SS
R = optional rewind

(RMD) UNIT, Lun, File, Key

SET, P(1), p(2),...,p(n)

A = addition/delete list
C = combine file output

E = end-of-file insertion

I = input source from PI

L = List output

N = new deck output

S = source record output

Vn = VORTEX option switch

Y = Copy option

- GANG,xxx
- DECK, deck name
- Comdeck, deck name, S
- Copy, first, last
- Random, first, last
- Append, first, last
- Edit, DEL/REN, deck name, new name,
new edition
- LIST
- CHECK
- INSERT (ADD), SE, NO
- REPLECE (DELETE, First, Last
- COMMON, name, action, first, last
name = name of the deck from the
currently open common file
action = INSERT or REPLACE

- COMSY, deck name, new name, new edition
- COMSY, deckname, editon, odate, update

- FILE (specify logical end-of-file)

- END, NN, directive

NN = optional number for /mem
directive = optional any JCP legal directive

SYSTEM MAINTENANCE

/SMAIN (JCP Directive)

SM**

Input Logical Unit

IN, lun, key, filename (old SGL)

Output Logical Unit

OUT, lun, key, filename (new SGL)

Input Logical Unit (new SGL)

ALT, lun, key, filename (new object)

Add to SGL

ADD, p(1), p(2),...,p(n)

p(1) is the name of an object module or
control record after which additions
are to be made.

Replace SGL

REP, p(1), p(2),...,p(n)

p(1) is the name of an object module or
control record that is to be replaced

Delete Record(s)
DE, recno 1, recno 2

Copy File(s)
FC, nfiles

Gang Load All Records
GA, (first, last), 'string'

List Records
LI, list
list A = list all out records
C = list only change records
N = suppress listing, except directives

Move Record(s)
MO, recno 1, recno 2, recno 3

Replace Record(s)
REPL, recno 1, recno 2

Rewind
REWI, p(1), p(2), p(3)
p(1) = IN, OUT, ALT

Write End of File
WE

Add String
SA, recno, (first, last), 'string'

Start Sequencing
SE, (first, last), initial, increment

Delete String
SD, recno, (n1, n2, n3)

Sequence Records
SE, (first, last), initial, increment

Stop Sequencing
SE, N

Replace String
SR, recno, (n1, n2, n3), 'string'

COMPRESSION/EDIT SYSTEM

/COMSY

(No RMD) ASSIGN, unit, Lun, R
unit = allowable SI, PI, BO, LO and SS
R = optional rewind

(RMD) UNIT, Lun, File, Key
SET, P(1), p(2), ..., p(n)

A = addition/delete list
C = combine file output

E = end-of-file insertion
I = input source from PI
L = List output
N = new deck output
S = source record output
Vn = VORTEX option switch
Y = Copy option

- GANG, xxx
- DECK, deck name
- Comdeck, deck name, S
- Copy, first, last
- Random, first, last
- Append, first, last
- Edit, DEL/REN, deck name, new name, new edition
- LIST
- CHECK
- INSERT (ADD), SE, NO
- REPLECE (DELETE, First, Last
- COMMON, name, action, first, last
name = name of the deck from the currently open common file
action = INSERT or REPLACE
- COMSY, deck name, new name, new edition
- COMSY, deckname, editon, odate, update
- FILE (specify logical end-of-file)
- END, NN, directive
NN = optional number for /mem
directive = optional any JCP legal directive

SYSTEM MAINTENANCE

/SMAIN (JCP Directive)

SM**

Input Logical Unit
IN, lun, key, filename (old SGL)

Ouput Logical Unit
OUT, lun, key, filename (new SGL)

Input Logical Unit (new SGL)
ALT, lun, key, filename (new object)

Add to SGL
ADD, p(1), p(2), ..., p(n)
p(1) is the name of an object module or control record after which additions are to be made.

Replace SGL
REP, p(1), p(2), ..., p(n)
p(1) is the name of an object module or control record that is to be replaced

Delete SGL
DEL, p(1), p(2), ..., p(n)
p(1) is the name of an object module or control record that is to be deleted.

List Old SGL
LIST

End of Input of SMAIN
END, L
L list old SGL

CONSOLE LOGGING

;SCHED, CONLOG, level, lun, key

ON { .AUT }
 { .MAN } .INI

OFF { .LOG1 }
 { .LOG2 }
 { .ALL }

PRINT { .LOG1 }
 { .LOG2 } .S
 { .ALL }

SWITCH

HELP

EXIT

STATUS

VORTEX LOGICAL UNIT ASSIGNMENTS

NUMBER	NAME	FUNCTION
0	DUM	FOR I/O SIMULATION
1	OC	OPERATOR COMMUNICATION DCONTROL
2	SI	JCP CONTROL
3	SO	SYSTEM OUTPUT MESSAGES
4	PI	INPUT FOR DASM, FOR, RPG (SOURCE INPUT)
5	LO	OUTPUT LISTING
6	BI	UNIT FOR OBJECT MODULE INPUT
7	BO	UNIT FOR OBJECT RECORD OUTPUT
8	SS	SYSTEM SCRATCH, PASSII SOURCE INPUT UNIT
9	GO	ADDITION COPY OF OBJECT TEST FOR LOAD & GO MODE
10	PO	COPY OF DASM SOURCE
11	DI	DEBUGGING INPUT
12	DO	DEBUGGING OUTPUT